TEST PLANNING APPROACH AND LESSONS*

Douglas A. Parkinson and Kendall K. Brown
TD51 Engine Systems Group
Marshall Space Flight Center
Huntsville, AL

ABSTRACT

As NASA began technology risk reduction activities and planning for the next generation launch vehicle under the Space Launch Initiative (SLI), now the Next Generation Launch Technology (NGLT) Program, a review of past large liquid rocket engine development programs was performed. The intent of the review was to identify any significant lessons from the development testing programs that could be applied to current and future engine development programs. Because the primary prototype engine in design at the time of this study was the Boeing-Rocketdyne RS-84, the study was slightly biased towards LOX/RP-1 liquid propellant engines. However, the significant lessons identified are universal. It is anticipated that these lessons will serve as a reference for test planning in the Engine Systems Group at Marshall Space Flight Center (MSFC).

Towards the end of F-1 and J-2 engine development testing, NASA/MSFC asked Rocketdyne to review those test programs. The result was a document titled, *Study to Accelerate Development by Test of a Rocket Engine (R-8099)*[1]. The "intent (of this study) is to apply this thinking and learning to more efficiently develop rocket engines to high reliability with improved cost effectiveness."[1] Additionally, several other engine programs were reviewed- such as SSME, NSTS, STME, MC-1, and RS-83- to support or refute the R-8099.

R-8099 revealed two primary lessons for test planning, which were supported by the other engine development programs. First, engine development programs can benefit from arranging the test program for engine system testing as early as feasible. The best test for determining environments is at the system level, the closest to the operational flight environment. Secondly, the component testing, which tends to be elaborate, should instead be geared towards reducing risk to enable system test. Technical risk can be reduced at the component level, but the design can only be truly verified and validated after engine system testing.

INTRODUCTION

Verification planning is critical throughout the design cycle because it forces the designers to understand and communicate the method in which the final product can be verified, validated, and later certified and accepted. The test planning approach presented in this paper focuses upon the testing portion of the verification process for full-scale development (FSD) programs involving liquid propellant rocket engines.

Unlike most other methods, verification by test tends to be costly to the program in three ways: (1) by incurring a significant amount of total development cost (20-30%), (2) requiring a significant amount of manpower (operational support for test facility and hardware in addition to engineering support), and (3) demanding a significant portion of total development time. For these reasons, program managers continue to look either at methods for reducing the cost of tests, or reducing the number of total tests all together.

In recent history, verification by analysis has become more en vogue as computer power has increased exponentially over the last few decades. Analysis has taken a much more active role in the verification process- much as it has in the design process. Computers have enabled computational fluid dynamics (CFD), 3-D modeling of engine components, and detailed stress and thermal calculations, allowing engineers to make much better "guesses" of the actual conditions of the engine. In testing, computer power enables prediction of test conditions and aids in translating test data. However, "the complexity of interactive characteristics of the propulsion, structural and electrical systems defies accurate analytical representation. Propulsion system testing provides the necessary test data for 'model basing' thus enhancing system analysis techniques."[2] This approach attempts to take into account the advancement of

*This effort was performed under NASA contract no. NAS8-01107

analysis tools by including a review of programs over the course of liquid rocket engine history, starting with programs as early as F-1 and concluding with programs as recent as RS-83.

Verification by analysis tends to be the most cost effective route, having only one test at the system level before delivery. Because the current analysis tools and methods still do not provide sufficient confidence to eliminate testing, a delicate balance for the two together, must fulfill the verification process. Obviously, the relative technology readiness level (TRL) of the enabling and enhancing technologies within the scope of a given development program play a significant role in the test plan development. The general rule applies- the lower the TRL, the more extensive the test and analysis programs. It is also understood that each program has unique goals and objectives. Therefore, this approach does not try to imply that all test plans should be identical. It does attempt to provide a common set of guidelines or a method, based on historical programs, for approaching test plans.

RESULTS AND DISCUSSION

An Approach

Given the boundary conditions set forth above, an approach must be general enough to be applicable to all programs, but still be specific enough as to guide the test planners into a unified direction. This approach is in no way finalized and will be refined as future experiences are documented. It is presented in the form of arguments, as this subject can be somewhat controversial. A significant amount of information is extracted from the references to support the arguments. As in any interpretation, information can be unintentionally mistranslated from the original author's intent.

I. *Argument 1: Engine development programs should adopt a risk-based test philosophy that focuses on early-as-feasible system testing*

This is the primary point of the approach and was best summarized, for all the sources reviewed, by the recommendations in *Advanced NSTS Propulsion System: Verification Study*[2]:

- "System level tests are necessary to verify end-to-end tolerances for timing, limit checks, and remedial actions. Incremental subsystem tests do not adequately provide the system response signature found in integrated systems tests.
- The complexity of interactive characteristics of the propulsion, structural and electrical systems defies accurate analytical representation. Propulsion system testing provides the necessary test data for 'model basing' thus enhancing system analysis techniques.
- Propulsion system testing determines hardware integrity and functional performance in the best possible environment excluding flight. Testing also certifies environments utilized for component development and qualification.
- Propulsion system testing integrates vehicle and ground hardware and procedure for propellant loading, safing, and firing operations for all systems.
- Propulsion system testing provides a resource for determining stage/engine design margins, developing procedures and timelines and confirmation of extrapolated criteria used in engine development.
- Demonstrate margin during subsystem and system level test"[2]

More detailed support for *Argument 1* is extracted from *R-8099*[1] by adding quantitative results from the F-1 and J-2 programs. The statistics reveal greater than 50 percent of the total failures being attributed to system interactions.

- "F-1 engine development program utilized a philosophy of minimum component developmental testing prior to incorporation into the system level. A review of the first 20 failure modes encountered on the F-1 engine system indicates that this was basically a good decision."[1]
  - 6 failures could have been detected at the component development level
  - 2 were failures associated with the MCC and gas generator
    - "These failures indicated that these components require preliminary development and may benefit from subsystem development when the cost and time of adequate subsystem facilities is practical with respect to the cost of the engine system test."[1]
  - 12 failures could not have been detected by component development programs
- An analysis conducted on the F-1 data suggests, "...that ½ of the failure modes experienced on the F-1 engine require engine test experience to detect, that is, no feasible lower level testing could be

expected to simulate the complex functional and environmental factors that produced the failure mode. The other ½ are those failure modes for which lower level tests could be expected to produce and are evenly divided between the subsystem and component test categories."[1]

- Based on the "qualification test summary for the F-1, J-2 and H-1 engines"[1]
  - 137 components subjected to component qualification test
  - 77 failed to meet test criteria
    - "51 of the 77 failure modes were not applicable to engine environment, as they (the components) were either defined as adequate based on engine testing or requalified at different (more relevant) test conditions…"[1] without redesign.
  - 8 of remaining 60 failed during engine testing
- † "An analysis of component qualification test results and the action taken on the results of those tests was conducted to determine how efficiently this type of component testing is at detecting design deficiencies. Component Qualification test results for the F-1, J-2, and H-1 engine programs were reviewed to determine failure frequency, applicability of failures to engine environment, and the corrective action taken. In addition, a search for failures encountered during engine test caused by design weaknesses in components which had satisfactorily passed component qualification was conducted.
  1. 50% of the components failed to pass component qualification
  2. 50% of the failures resulted in test requirements modification
  3. 30% of the failures resulted in redesign
  4. 20% of the failures were deemed not applicable to the normal operating environment
  5. 70% of the failures did not reflect a similar failure mode in engine test
  6. 30% of the failures did reflect a comparable mode in system level test
  7. 13% of the successes exhibited a failure mode in system test that was not detected in component qualification"[1]

Less than a decade later, the Space Shuttle Main Engine (SSME) Program reached a similar conclusion. The SSME program migrated from a more heavily component-oriented test program to more system-oriented test program.

- "Within the program realignment of 1974, it was decided that the first article of each major component would be allocated to the ISTB (Integrated Subsystem Test Bed). This action would accelerate engine testing and the discovery of any potential major ~~system~~ problems, but would delay the beginning of the component test program until after the second article had been assembled."[3]
- "The component test program, if pursued as originally planned, would have drained the valuable resources from the engine test program to develop the complicated test facilities. The NASA administrator, Dr. Robert Frosch, stated in testimony to the Senate Subcommittee on Science, Technology and Space, that '…we have found that the best and truest test bed for all major components, and especially turbopumps, is the engine itself.' Largely due to the lack of sufficient resources to pursue an aggressive component test program in addition to the engine test program, the Coca area test facilities were gradually phased out from November 1976 to September 1977."[3]
- *Entry 960* of the *NASA Lessons Learned Database* states, "the use of an Integrated Subsystem Test Bed and Full Engine Testing in preference to individual component testing speeds up the development process and produces a high reliability engine that takes into account subsystem interactions."[4]

The issue is still remains after 30 years as the Rocketdyne RS-83 Risk Advisory Board (RAB) debated the test planning approach for the project. The RAB recommended changing the baseline to include component and system level testing, eliminating the "powerpack" from the test program. This path would include a slight increase in component tests, but improved the potential for early risk retirement and increased knowledge geared towards engine system test. The downside would be that "powerpack" would address some of the system-level interactions that component-level testing is not able to simulate. However, the readiness level of the component test stands is significantly greater than the "powerpack" stand and the component test stands are also significantly less complex. Additionally, the elimination of "powerpack" testing moved the engine system testing earlier in the program.

In all cases (NSTS study, R-8099, SSME, and RS-83), the importance of integrated, systems-level testing is emphasized. It is also important to note that none of the references exclude component testing. *Argument 2* focuses on the need for component testing.

† There are multiple ways to interpret this information (seems incomplete), but it is believed that the main point of this paragraph and supporting statistics supports Argument 1

II. *Argument 2: Component testing should not be extremely elaborate, but instead be geared towards reducing risk to enable the system level test program*

It is understood that component testing must occur to a certain extent before integrating into the system. TRL for the technologies, which are based on the scope of the program, that make up the components will have significant effect on their individual risk levels. Component test plans should be based on retiring risks that allow the component to become *functional* to proceed to system level testing.

- "Laboratory test of critical components are performed as soon as possible to ensure that the component will operate in the system... However, component tests have the following drawbacks:
  - Actual component environment and input/output requirements can be determined only by engine test.
  - Many complex environmental laboratory facilities must be built and operated to test each component to its assigned or measured requirements.
  - Even with a massive effort in areas 1 and 2 above, it will be impossible to simulate all the known environments. The unsuspected phenomena that system testing gradually reveals will, of course, not be duplicated."[1]
- With respect to component testing, considerable time was spent trying to verify specification requirements that far exceeded the true environmental conditions.
  - Example: "...designing components to absorb much higher than normal heat loads ... where estimates of the temperatures varied from –300 to +400 F and in reality were found to vary between 0 and –100 F."[1]
- "The exceptions are the gas generator and main injector, the thrust chamber assembly, and the turbopump assembly."[1]
  - "The turbopump assembly development test period was short... and is considered a functional checkout rather than a development program. However, it did serve to verify turbopump operational capability, and failure to operate satisfactorily would have delayed initiation of engine test."[1]
  - "The criticality of the turbopump to the engine operation dictates that the turbopump assembly be subjected to sufficient subsystem testing to verify integrity at the nominal operation point."[1]
- "For all other components (no specifics in the literature, but likely referring to valves, lines/ducts, nozzle, controller, etc),... a general objective to extensively laboratory test components prior to engine testing would not be an efficient development method."[1]
- "Neither of the J-2 turbopumps was tested as a component prior to engine experience but both would have benefited from it."[1]
  - "Under component design, the fuel turbopump, as expected, accounts for the majority of the cutoffs, followed by the oxidizer turbopump. It is significant to note that neither of these two components had any degree of separate pit testing prior to engine testing because of time limitations. Both could have significantly benefited from some prior development."[1]
- "The J-2 gas generator design received and required more than a full year of development time prior to engine test... Both component and early engine test appear to be necessary to detect problems on this particular component."[1]
- "Of course, when a particular component failure mode is experienced in system testing, the advantages of exploring variable influences or overstress in the laboratory will be considered."[1]
- "An engine should be assembled as soon as possible. This is the only way to assure that components are operating in the correct environment, and to do so reveal the system type of failures. Preliminary component testing is probably restricted to early tests to help the designer, and to assuring that the component has reasonable chance of operation on the engine."[1]
- "This study indicates that a development program must remain heavily engine test oriented. It is difficult to simulate engine environment from laboratory component testing. The typical Phase 2 environment related problems are especially difficult to measure."[1]

These points, joined by those supporting *Argument 1*, convey a much more complete idea for a test planning approach. System-level testing is the desired condition, and component testing should be limited to confidence builders (functional checkouts) as much as possible. Spending extreme amounts of time fully characterizing the components many times is unnecessary as test conditions generated before system-level testing may be too extreme, or worse, not extreme enough. However, components like turbomachinery and combustion devices do need characterization through component-level testing because of their relative complexity. It should be noted that component-level testing can be an effective trouble-shooting method if the facilities can sufficiently recreate, with a degree of accuracy, the system environment.

The complexity for test planning has just exponentially increased. The numerous components each have functional requirements that demand component-level tests to ensure functionality at the system-level test. A large number of test objectives exist for the entire system, which encompasses TRL-increasing development tests. The last argument suggests a method in which to construct a test plan with all the tiers of test objectives in mind.

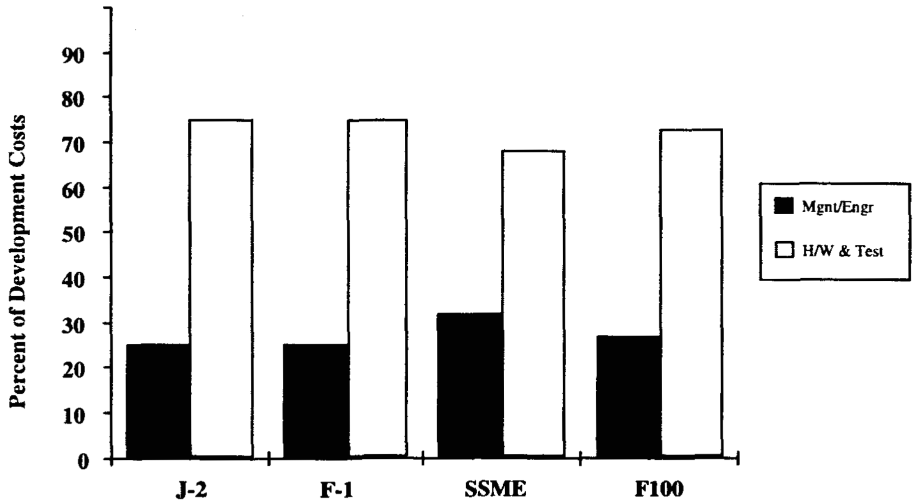
III. Argument 3: Multiple objectives per test minimizes the number of tests

The obvious goal is to minimize cost to the program by constructing a test plan that minimizes the necessary number of tests to accomplish all the objectives. Jan Monk, author of *Reusable Launch Vehicle Main Engine Development*[5], suggests that statistical design of experiments (DoE) and similar methods may minimize test cost for a development program.

Monk documents a method (based on DoE) for optimizing the number of tests that allows tracking which factors drive the number of tests up and down during development.

- "Number of test exposures required defines the number of test and amount of hardware required"[5]
- It states that the number of tests is a product of the requirements, number of design cycles, and variability in the hardware where:
  - *Requirements* = minimum required exposures to verify requirements can be achieved by using Design of Experiments (DoE) techniques.
  - *No of Design Cycles* = f1(changes in top level requirements) + f2(accuracy of derived requirements) + f3(accuracy of analyses) + f4(accuracy of fabrication) + f5(design time) + f6(fabrication time) + f7(design)
  - *Variations in Hardware* = f8(fabrication process variability) + f9(material properties variability) + f10(accuracy of "computed loads")"[5]

Monk also recommends targeting hardware and test costs because "reducing engineering and management costs by 50% only reduces development costs by 12-15%,"[5] as illustrated in Figure 1.



Reducing Engineering and Management Costs By 50 Percent
ONLY Reduces Development Costs by 12-15 Percent

TARGET Hardware/Test Costs

Figure 1[5]

HISTORICAL FSD COST DISTRIBUTION

History shows major cost elements are consistent

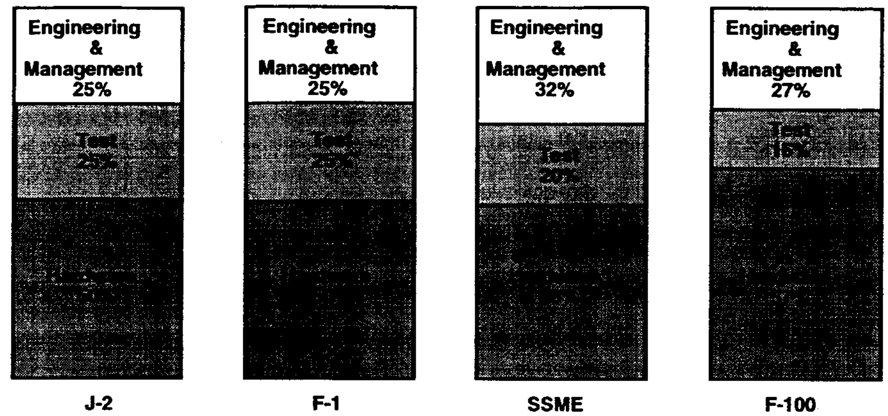


Figure 2[6]

Figure 2 supports Monk's argument as well. The Rocket Engine Development Team (REDT), a NASA/MSFC, Aerojet, Pratt & Whitney, and Rocketdyne consortium assembled in the early 1990's, reviewed historical engine development programs and support the DoE path in *Engine Development at Reduced Cost, Schedule and Risk*[6].

- "Minimize Test Costs
  - Limit test scope via minimum 'performance' requirements and elimination of technology stretch
  - Define objectives by failure mode and sensitivity analyses
  - Utilize design of experiments techniques for fewest tests
  - Optimize test levels (rig vs. component vs. engine) to least total test cost
  - Combine test objectives for fewest tests"[6]
- "Minimize Testing
  - Risk: Multiple objective testing may mask deleterious effects
  - Benefit: Overall cost and schedule of test phase is minimized"[6]

It is significant to note that although it is generally good to minimize the required amount of testing, it may mask unforeseen problems. For example, too many objectives for a test may be unachievable by the facility without significant additional cost. Another example may be life testing, which requires high time or longer duration tests, where fewer tests will not necessarily achieve this goal. Because of these examples and others like it, a list of lessons accompanies the approach to add some more detailed guidance.

Relevant Lessons

The arguments presented above guide the test planner in a relatively general direction. This section summarizes several lessons to increase the fidelity and establish a "best practices" in regards to test planning. It should also be understood that this list does not necessarily capture all the lessons (which are in no particular order), and may be modified as time progresses.

I. *Lesson: Test planning should be started early in engine development*

- "Ensure rigorous test planning is conducted early enough in the project to adequately scope the project. This will enable optimized resource allocation and minimize schedule impacts."[7]

II. *Lesson: Dedicate a significant portion of testing to engine environments characterization (nominal and off-nominal operation)*

- "Definition and test of engine operating environment early in the program will greatly accelerate the engine development."[1]

- "When the engine operating environment is unknown, such as in the beginning of a new engine program, it is recommended that a portion of the initial testing be conducted in the form of limits tests to map the engine vibration and stress environment by pursuing extensive accelerometer and strain gage instrumentation. First tests are at nominal or reduced values until it becomes reasonably operable. Limits are carefully probed as soon thereafter as possible. Test programs should be devoted to extensive limits tests, with corresponding increased rate of uncovering problem areas."[1]
- "It (SSME development program) also pursues engine hot fire tests that demonstrate the limits of the engine operational parameters and margins or over-stress tests to verify the full engine capability. Further, margin tests should be conducted to the extent that they reasonably represent potential engine operation in a degraded state."[4]

III. <u>Lesson</u>: *Test planning must take into account all verification activities- inspections, analyses, etc.*
- "Make detailed test development plans. The fewer tests permitted in the program permit, and require proportionally more time to be spent in planning and analysis of each test, and the teardown inspections of used hardware."[1]

IV. <u>Lesson</u>: *Test planning should include a significant amount of short duration tests to understand engine transients*
- "Minimum problems occur during mainstage. Therefore, limited full-duration tests are required during R&D testing. Greater effort again is required to resolve cutoff-oriented problems. Full-duration tests are not required to resolve these problems."[1]
- "Many short duration tests can be used to resolve all problems in this area (start transient)."[1]
- "...performance of engine at all corners of the operating box can be established prior to full-duration testing."[1]

V. <u>Lesson</u>: *Contingencies should be allowed for anomaly resolution, redesign incorporation, and retest/reverify*
- "Caution should be exercised to fully analyze and reverify the total potential effects that an "add on" feature or a redesigned part may have on interfacing parts. Often influences not outwardly obvious can only be detected by repeating prior design analyses and verification test programs assumed to be still valid"[1]
- Factors which have a significant influence on the time required to detect and/or correct failure modes are:
    - Hardware lead times- "The calendar lead time to procure material and fabricate typical F-1 engine components for development testing...is particularly significant relative to late definition of engine requirements imposed by stage interfacing considerations and can be a major contributing factor for late exposure of applicable hardware to engine testing..."[1]
    - "Extent of development testing required to verify a particular design feature"- "In addition to the types of tests required, a basic question is the number of hardware samples that should be exposed to the various types of development tests."[1]
- "Based on 152 F-1 engine design changes evaluated on engine tests during the F-1 Program"[1]
    - 121 have been satisfactory corrected
    - 31 failed to correct a prior mode or introduced a new failure mode as a result of the redesign
- "The average time to correct a failure mode on the F-1 engine, as measured from the first occurrence of the failure mode, was nine months. Categorization of typical correction times by cause of mode (Figure 2-24) indicates that 30 of the 35 failure modes are included in cause categories which average nine months or less to correct. The remaining five modes are associated with failures which are not as severe as the others, that is, the frequency of occurrence of the mode or the probability of significant operational impact is low. This indicates that failure mode correction priority was properly administered in the course of the F-l development program.
    - Eight of the 35 failure modes required longer than 12 months to correct."[1]

VI. <u>Lesson</u>: *Minimize program technology "stretches"*
- Technology "stretches" refers to technologies that have a relatively low TRL and need significant testing and characterization to bring it to a usable TRL level for implementation
- "The obvious conclusion is that major cost (and schedule) reductions are possible if hardware is reduced. This is particularly true if the hardware is both low in cost and mature to the extent that relatively few failures occur. The reduction in failures will also allow significantly fewer tests to be conducted, further reducing cost and schedule for FSD."[6]

VII. <u>Lesson</u>: *Liquid rocket engine development exhibit similar learning curves*

- "All prior engine (Titan II, H-1, RL-10, J-2, F-1, SSME) exhibit similar learning curves, with a level of maturity defined by a significant reduction in failure rates
    - o Failure distribution with test duration consistent
    - o Test type (e.g. nominal conditions vs. margin tests) affect the learning curve
    - o Learning curves suggest an evolution in rocket engines"[8]
- For SSME:
    - o 400 test learning curve on block I engine
    - o 200 addition test to acceptable level of maturity on block II to demonstrate reliability goal

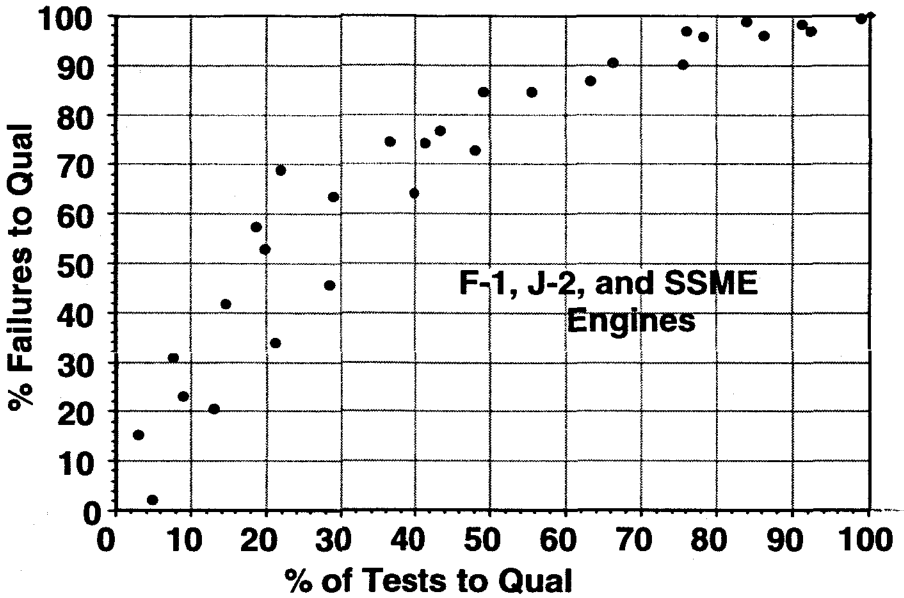


Figure 3[9]

VIII. <u>Lesson</u>: *Test planning should not expect a high test rate during development*

- Seen as relatively high rates for testing,
    - o "SSME average test rate of 2 tests/week/test position
        - ▪ SSME maintained a rest rate of 3 tests/week for a short time
        - ▪ SSME average 1.3 tests/week for a extended period"[8]
    - o "J-2 average 2.5 tests/week over 5 years"[8]

These lessons may overlap and may even be contradictory. A delicate balance must be worked out between the technical and programmatic team members in order to achieve the objectives of the program. Priorities must be set as to which requirements are critical and which can be compromised.

<u>Caveats</u>

A particular engine is made to accomplish a specific set of missions. Like any industry, there may be multiple solutions to satisfy the given program needs. This section addresses some of the possible caveats that can exist because of a particular cycle choice, and the possible effects of adding an additional development cycle (i.e. prototype) to the program prior to full-scale development (FSD). Much of this information is already embedded within the approach and lessons because the nature of the programs that were reviewed.

Other significant caveats exist such as propellant choices (i.e. RP-1, H_2, O_2) and intended use (i.e. booster, upper stage, expendable, reusable, man-rated). All play a role in test planning. Unfortunately, not all of them will be addressed within this paper, but are mentioned because they are believed to be factors in effective test planning.

I. <u>Caveat</u>: *Addition of the prototype*

Technological challenges are inherent to prototype design cycles, which is a primary reason for prototype programs. Technology challenges, seen and tracked as risks, tend to be significant in both number and their low technology readiness level (TRL). The test planning approach to the risk-adverse prototype design cycle can vary significantly from the follow-on FSD design cycle.

The expectation for prototype should be to include a significant amount of lower-level testing focused upon increasing TRL levels and reducing risk. Low TRL inherently brings more risk to the program and implies low confidence in the initial design. Therefore, it should be the expectation of the program to experience failure modes that may not be relevant to the nominal operation of the component. Lack of knowledge in the environments, facility complexity, TRL, or any combination of these can cause the testing to be misdirected into "chasing ghosts." Another significant issue may be that the testing proves the technology to be invalid for the intended application, which can open up an entire new line of testing. Significant contingencies should be allotted for the low TRL testing to truly characterize the technology. An over exaggerated verification program should be an expectation for prototype.

On the other hand, a prototype design cycle can greatly benefit the full scale development program by characterizing the environment and design analysis methodologies, significantly decreasing FSD risk. FSD System-level testing should be achievable much earlier because of the ability to characterize components more rapidly, and with greater confidence. This is where the addition of a prototype design cycle can ultimately realize cost and schedule savings for the FSD program- critical processes will have been established and detailed requirements and verification plans can be constructed relatively quickly. Similarly, fewer design changes can be expected, fewer failures, and less wasted testing ("chasing ghosts"). Essentially, the technology development would be more independent from product development.

Therefore, a prototype test plan should be more elaborate. It could contain many more tiers for testing based upon the risks to the program. All scoped technology challenges should have significant traceability to FSD, but should have sufficient back-ups to ensure program success. These back-ups may need to be included and carried in an alternate test plan to be properly mitigated.

II. <u>Caveat</u>: *Cycle variations*

Another major difference between programs may be the engine cycle. The basic approach can be applied to all engine cycles; however, the cycles do differ and require different levels of testing in different areas. Although many engine cycles exist, only the three most common cycles- gas generator (GG), staged-combustion (SC), and expander (EX)- are addressed.

Regarding the gas generator (GG) cycle, engine programs tend to emphasize the relative simplicity of the cycle and should base a test plan that simplicity. Extensive component testing does not necessarily make sense unless the engine environment has been sufficiently characterized. The test plan should proceed to engine system testing early after component functional tests. If need be, component testing can be continued in more detail to tackle trouble areas and redesigns. It may not be cost effective to add a prototype development cycle unless there are significant technology challenges. In many cases, "battleship" or non-flight hardware can be used and refined during FSD.

Staged-combustion (SC) cycle engine programs tend to be more heavily component loaded since the relative complexity is high. Complexity drives the test plans towards increased functional characterization of components (relative to a GG) prior to system integration. However, the early system testing approach is still valid. Understanding engine environments and transient operation is necessary to validate the component-level characterization. The addition of a prototype cycle would benefit the program in this case due to the complexity of the cycle and its operation.

The last to be addressed is the expander (EX) cycle. This cycle is relatively simple because it uses heat gained from the nozzle to drive the turbomachinery. This eliminates the need for a gas generator or preburner, simplifying the cycle. A serious limitation is size which tends to be limited by the heat transfer capabilities of modern materials and heat absorption of coolant fluids. There are many variations of this cycle with varying complexity. An open EX (a.k.a. coolant bleed) cycle is similar to a GG cycle minus the gas generator. The closed EX is much like a SC cycle without the preburner. Again, complexity would be a driver in the component-to-system test ratio. Because the heat transfer drives the cycle, the lower thrust classes (below 60 klbf of thrust) are much more technically feasible with the current state of technology. The higher thrust classes (above 100 klbf of thrust) are much more difficult and have little to no experience base. A prototype would be beneficial here to increase the TRL of the enabling technologies. The early system test is applicable here due to the highly integrated nature of the nozzle coolant to the turbomachinery drive.

The general approach of early-as-feasible system test seems to apply regardless of cycle. However, inherent complexities of each cycle will draw the test plans to be significantly different.

SUMMARY AND CONCLUSION

"The devil is in the details." And, at least for test planning, this still remains to be true. There seems to be countless variables that have to be taken into account when constructing a test plan. Engine cycle complexity and number of complete design cycles can have major affects on cost, schedule, and risk. All of these details need to be taken into account when determining the proper number and which kind of component, subsystem, and system tests to incorporate to the plan.

The set of lessons reiterated (reiterated because they are not new, but need revisiting) seem to be valid for all programs present and future. Particular emphasis should be put on environment characterization and transient refinement; areas which still tend to be very costly to engine development.

Concentration on environments characterization and transients can help set the framework for a component and system test plan. The goal should be to strive for early-as-feasible, system-level testing to fully characterize system interactions and environments; where historically, lack of knowledge in these areas can be held accountable for better than half of the failures experienced during full-scale development.

REFERENCES

[1] Aldrich, D.E. R-8099 Study to Accelerate Development by Test of a Rocket Engine. 12 Dec.1969.

[2] Yeager, Jim and Charles Wood. "Advanced NSTS Propulsion System: Verification Study," July 1989

[3] Biggs, Robert. Space Shuttle Main Engine: The First Ten Years. Liquid Rocket Propulsion History Colloquium, 2 November 1989.

[4] NASA Lessons Learned/Best Practices Database. NASA Technical Standards Website

[5] Monk, Jan C. "Reusable Launch Vehicle Main Engine Development," March 1996

[6] Rocket Engine Development Team. "Engine Development at Reduced Cost, Schedule and Risk," June 1993.

[7] MC-1 Project Office. MC-1 Engine Project Termination Closeout Report, MSFC-RPT-3195, December 2001.

[8] Lyles, Gary. "STME Development Plan, Historical Perspective," May 1991.

[9] Lyles, Gary and Jerry Redus. "Lessons Learned from Saturn and Shuttle Boosters," 1993 AIAA Joint Propulsion Conference Short Course, June 1993